A Silicon Anti-Virus Engine

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

Adrian Tang Dr. John Demme Prof. Simha Sethumadhavan Prof. Salvatore Stolfo

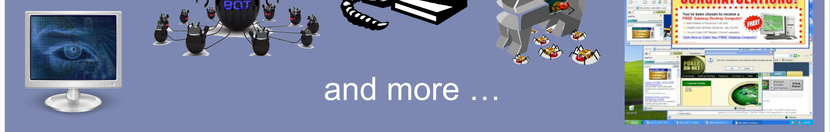
Motivation

- Proliferation of malware – stealthier and increasing in number
- Software-level detection mechanisms have limited effectiveness

Rethinking malware detection with hardware approach and low-level features

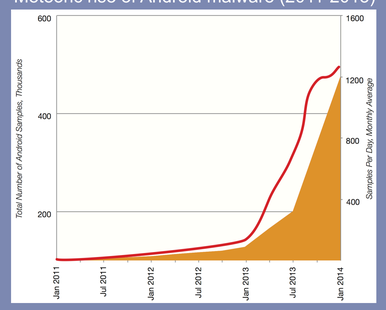
Growing Malware Threats

Botnet Worms Trojans Spyware Adware



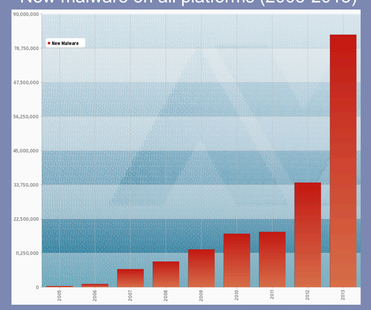
and more …

Meteoric rise of Android malware (2011-2013)



Source: Fortinet (2014)

New malware on all platforms (2005-2013)

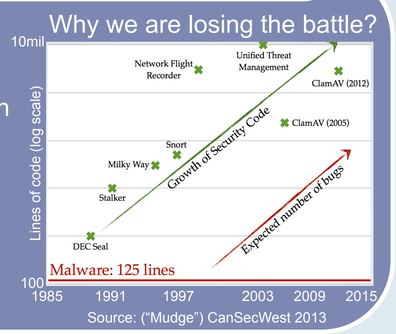


Source: AV-Test (2014)

Limitations of Software Anti-Virus

- Same level as software malware
  - Prone to attacks/subversion
- Complex software implementation (many lines of code)
  - High bug density
- Signatures typically use static characteristics of malware
  - Static analysis can be defeated with trivial variants

Why we are losing the battle?

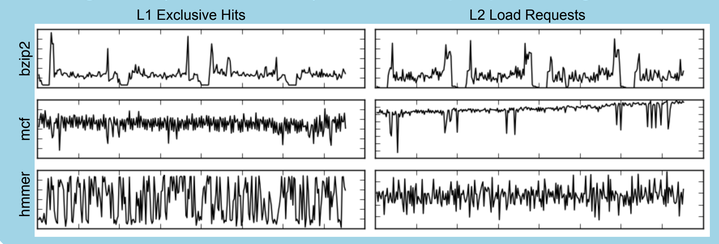


Source: ("Mudge") CanSecWest 2013

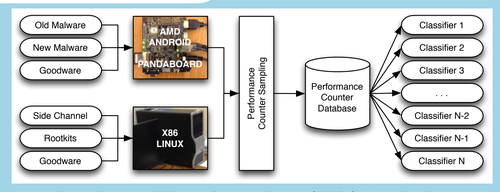
Catching Seen Malware [1]

Insight

Programs (and malware) exhibit unique µArch signatures.



Methodology

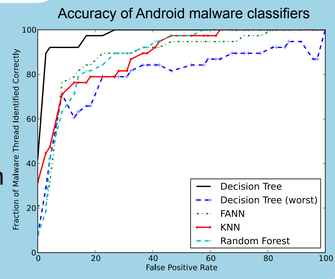


- Used supervised machine learning (ML) techniques to train models to characterize dynamic behavior of
  - 503 Android malware apps
  - 210 Android benign apps from Google Play
- Evaluated classifiers with different variants in the same malware family
- Also explored feasibility with Linux rootkits and cache side-channel attacks

Detection Results

- Android malware
  - 82.3% accuracy
- Linux rootkit
  - 60% accuracy
  - Difficult problem; rootkits are tiny slices of execution
- Side-channel attack
  - 100% accuracy; No false positive

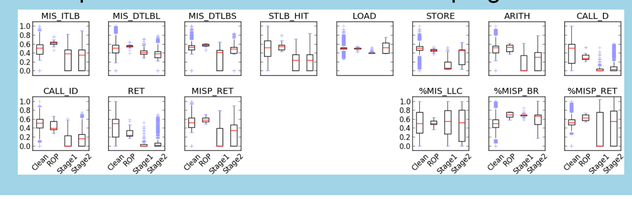
Accuracy of Android malware classifiers



Catching Unseen Malware [2]

Insight

Malware shellcode execution causes deviations in baseline µArch and arch characteristics of programs.

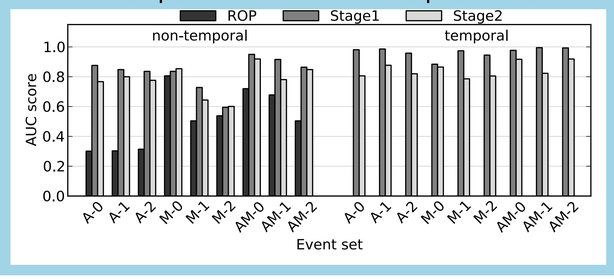


Methodology

- Used unsupervised ML technique (One-Class SVM with RBF kernel) to train baseline dynamic behavior models for
  - Internet Explorer 8
  - Adobe PDF Reader 9
- Evaluated detection models with *Metasploit*- generated exploit variants
  - Target IE, Flash plugin, PDF plugin/standalone versions
  - Multi-stage exploit process (*ROP* → *Stage1* shellcode → Stage2 payload)
- Different feature extraction methods (temporal vs non-temporal models)

Detection Results

- 99.5% AUC score for AM-1 event set (`STORE, LOAD, MISP_RET, CALL_ID`) for detection of Stage1 shellcode
  - 1.5% slowdown with sampling granularity of 512k ins.
  - 100% true positive with 1.1% false positive rate



[1] John Demme, Matthew Maycock, Jared Schmitz, Adrian Tang, Adam Waksman, Simha Sethumadhavan, and Salvatore Stolfo. 2013. "On the feasibility of online malware detection with performance counters." In Proceedings of the 40th Annual International Symposium on Computer Architecture (ISCA '13). ACM, New York, NY, USA, 559-570.

[2] Adrian Tang, Simha Sethumadhavan, and Salvatore J. Stolfo. "Unsupervised anomaly-based malware detection using hardware features." In Research in Attacks, Intrusions and Defenses, pp. 109-129. Springer International Publishing, 2014.